WEST Search History

DATE: Thursday, February 07, 2002

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L10	l1 and l9	13	L10
L9	l7 or l8	319	L9
L8	(((710/21)!.CCLS.))	153	L8
L7	((710/20)!.CCLS.)	196	L7
L6	l1 same l2	4	L6
L5	l1 same l3	66	L5
L4	l1 and l3	980	L4
L3	(concurrent$ or simultaneous$) near5 process$	69665	L3
L2	(concurrent$ or simultaneous$) near5 (task$ or job$) near5 process$	912	L2
L1	((releas$ or free$) near5 (server or processor or microprocessor))	9288	L1

END OF SEARCH HISTORY

WEST

Generate Collection Print

L6: Entry 2 of 4 File: USPT Aug 25, 1987

DOCUMENT-IDENTIFIER: US 4689764 A
TITLE: Method and apparatus for formatting a line of text containing a complex character prior to text justification

Brief Summary Paragraph Right (4):
Word or text processing systems have been developed for formatting alphanumeric data into a suitable format for specified printed documents, e.g., letters, reports, text books, magazines and newspapers. In previous word processing systems, substantially all of the formatting was accomplished in the word processor. A word processor normally comprises a keyboard entry display terminal, storage means, and a central processor. The operator utilizing the display terminal arranges the information into a preselected alphanumeric line and page format. The system then transmits the alphanumeric data, together with the formatting data, to the printer for outputting the previously formatted document. More recent word or text processing systems have sought new approaches to increase the output of the system. Improved word processing systems have included formatting capability in the printer so that the word processor would be released to carry out further word or text processing activities as the foreground task, while the printer could function simultaneously to complete its formatting functions as a background task.

WEST

Generate Collection Print

L10: Entry 7 of 13 File: USPT Jun 17, 1997

DOCUMENT-IDENTIFIER: US 5640592 A
TITLE: System for transferring utility algorithm stored within a peripheral device to a host computer in a format compatible with the type of the host computer

Detailed Description Paragraph Right (24):
As an alternative to downloading an algorithm to the personal computer from the disk drive assembly, the present invention also contemplates a second embodiment which executes the algorithm directly within the disk drive assembly, thus eliminating the need to download the algorithm and freeing up the processor of the host computer to perform other tasks instead of executing the utility algorithm. Additionally, executing a utility algorithm within the peripheral device instead of by the host computer does not require the algorithm to be executed by different microprocessors running different types of operating systems. This will simplify the requirements for the algorithm when the peripheral device is to be used with different types of computers, as compared to the first embodiment, as the algorithm will only execute on one type of processor or processing circuitry which is located within the peripheral device.

Detailed Description Paragraph Right (27):
The specific structure of the disk drive assembly is not limited to the illustration of FIG. 4 but any type of circuitry within a peripheral device may be utilized as long as the circuitry is capable of executing the desired algorithm, thus freeing up the host processor 250 to perform other tasks. When flexibility is a desired feature for the functions of the disk drive assembly, the nonvolatile memory 212 and/or processing circuitry 230 will be contained on a device which is easily connectable to and removable from the disk drive and contained on a plug-in device such as a PCMCIA card. Even though the algorithms described with respect to FIG. 4 are described as running within the disk drive assembly, a variation of the invention includes the host computer controlling the execution of the algorithms by providing start commands, stop commands, and controlling the various parameters of the algorithms.

Current US Cross Reference Classification (1):
710/20

L10: Entry 11 of 13 File: USPT Dec 3, 1991

DOCUMENT-IDENTIFIER: US 5070477 A
TITLE: Port adapter system including a controller for switching channels upon encountering a wait period of data transfer

Brief Summary Paragraph Right (5):
Attempts have been made to free the central processor from this I/O execution so that the central processor can spend more time on user jobs by supplying a separate general purpose processor to operate independently in the control of input/output data transfers. However, there must be some communication between the two processors in order to assure that the data required by the main central processor is received in its main memory prior to the central processor utilizing that data.

Brief Summary Paragraph Right (11):
In order to accomplish the above-identified objects, the present invention resides in a port adapter or bus driver for an input/output system for a large data processing system. The port adapter is coupled to an I/O processor of that system and also to the main memory of the system so that when the port adapter is selected by a system interrupt message from the I/O processor, it can begin its data transmission between a selected peripheral device and main memory without further assistance from the I/O processor or other processors in the system. A task control processor is provided with the I/O system for the scheduling of different central processors for the highest priority processes to be run. When an I/O operation is detected, the respective central processor is released from that process that it is currently running so that it can be assigned to the next highest priority process. When the requested I/O operation has been completed, the Task Control Processor is signaled so that the task control processor can put the requesting process back into the priority list of processes to be run by the main central processors.

Detailed Description Paragraph Right (4):
As has been indicated above, the function of the present invention is to relieve the operating systems and the respective central processor 10, which execute those operating systems, of all I/O operations so that central processors 10 will have more time for the execution of user jobs. When a given central processor 10 is executing a process from one of memory modules 12 and encounters an I/O operation, the corresponding I/O control block is created and the I/O instruction is sent to I/O system 13 by way of memory controller 11 and the processor 10 is released to begin executing the next highest order process from one of memory modules 12. When the I/O operation has been completed, the requesting process is then rescheduled in a priority list of processes for further execution by the next available central processor 10.

Current US Original Classification (1):
710/21

WEST

Generate Collection Print

L10: Entry 12 of 13 File: USPT May 7, 1985

DOCUMENT-IDENTIFIER: US 4516199 A
TITLE: Data processing system

Detailed Description Paragraph Right (15):
With respect to the high speed devices, the actual control of the information transfer to or from the main store is done via the direct memory access controller 30. This control is initiated, monitored and terminated by the auxiliary processor 16 via the input-output bus connection to the direct memory access controller 30 and to the device itself. The auxiliary processor 16 sends control information and receives status information from the device controller. If desired, a high speed device can be operated in the system without a direct memory access controller 30. In this case, the direct memory access junction is provided by the auxiliary processor. The actual transfer of data, using control provided by the direct memory access controller 30, does not load the auxiliary processor 16 thus freeing it for other tasks. The hardware of the direct memory access controller 30 has access priority to the main store bus 14 over any auxiliary processor 16 or execution processor 18, and it shares the main store bus 14 in a priority protocol with other direct memory access controllers.

Detailed Description Paragraph Right (55):
The input-output system fulfills some desirable objectives such as relief or input-output computations and testing by the execution processors 18, use of simple and low cost device controllers, the special high speed burst capability and data path multiplexing for efficiency and hardware failure handling. The input-output system has a redundant parallel design in that a partial failure may be handled by the remaining components. As previously mentioned, the auxiliary processors 16 are programmed with nano-instructions to allow them to do the testing and execution of input-output transfer thus freeing the execution processor 18 or central processing unit from the steps of input-output communications. The auxiliary processors 16 are configured on multiple input-output busses, thus providing multiple data channels, and the particular channel is determined by the supervisory processor 20 which is the ultimate controller in the system. On the channel the data exchanges are controlled by the intelligence of the auxiliary processor 16. The peripheral device controllers are controlled by the micro-programs in an auxiliary processor 16.

Detailed Description Paragraph Right (73):
While the above described procedure is the preferred way of handling data flowing in relatively low speed, it is in the nature of this invention to support very high speed data rates without the need to tie up an auxiliary processor 16. This is accomplished by using the DMA (Direct Memory Access) controllers 30. The auxiliary processor 16 is used to establish logical connections between a high speed device controller, say 108, and a DMA controller, say 30. Also, the auxiliary processor 16 is used to set up all control information for the transfer, as will be described, and to monitor end of transfer conditions. While the actual transfer is performed, the auxiliary processor is free to communicate with other devices.

Detailed Description Paragraph Right (76):
Once the direct memory access transfer is initiated, both the input and output bus and the auxiliary processor 16 are free. The DMA controller 30 will feed the device 108 continuously, will buffer the data in its own memory, and will request the memory bus 14 each time a physical word, 64 or 72 bits, is available until the byte count is reduced to zero. At that point in time, the bytes which do not constitute a full word are transferred to the main memory. The data flow from the device 114 will not be interrupted but continues and is buffered in the memory of the direct memory access controller 30. If there is an interrupt, the auxiliary processor for that controller, in particular auxiliary processor 16, is interrupted indicating a status change in the

DMA controller 30. At that time, the auxiliary processor 16 may stop the flow of data into the DMA controller 30 and void the first-in first-out mode of operation or it may reload the address and count registers of the DMA controller 30 with new values and direct it to transfer the data buffered in it to the next location therein or it may reload the controller with new commands and initiates a new input-output sequence.

Current US Original Classification (1):
710/20